**ESCUELA POLITÉCNICA NACIONAL**

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**BASES DE DATOS (TSDS)**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | Bases de Datos |
| PROFESOR:  FECHA DE ENTREGA: | Ing. Lorena Chulde  2025 - 02-05 |
| PERÍODO ACADÉMICO: | 2024-B |
|  | |

**TALLER**

Grupal

|  |
| --- |
|  |
| **TÍTULO**  **PROYECTO FINAL** |
|  |
| Qué es la integración de bases de datos ... |
|  |
| **Estudiantes**  **Luis Adrian Ramos Guzmán**  **Josué Eduard Guerra Lovato**  **Henry Tonato** |

**1. Resumen Ejecutivo**

El proyecto consiste en el diseño y desarrollo de un sistema de gestión de bases de datos para una aerolínea exclusiva de vuelos directos, denominada Valvoline. Los objetivos principales son optimizar la gestión de vuelos, pasajeros, tripulación y mantenimiento de aviones, garantizando la seguridad y eficiencia operativa. Las tecnologías utilizadas incluyen MySQL para la gestión de la base de datos y diversas herramientas de modelado de datos.

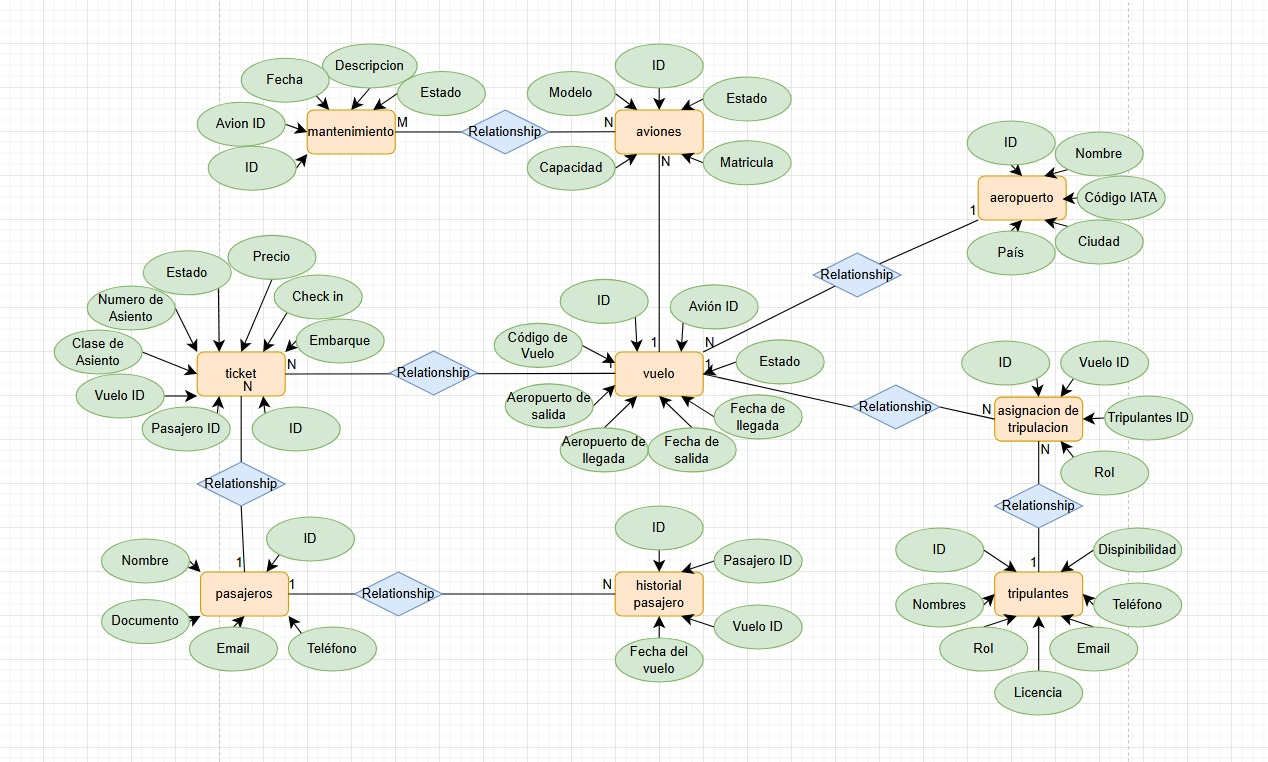
**2. Introducción**

La gestión eficiente de bases de datos es crucial para una aerolínea, ya que permite manejar grandes volúmenes de información de manera organizada y segura. Este proyecto tiene como propósito diseñar una base de datos robusta que facilite la administración de vuelos, pasajeros, tripulación y mantenimiento, mejorando así la operatividad y la toma de decisiones.

**3. Modelado de Base de Datos**

**3.1 Modelo Conceptual**

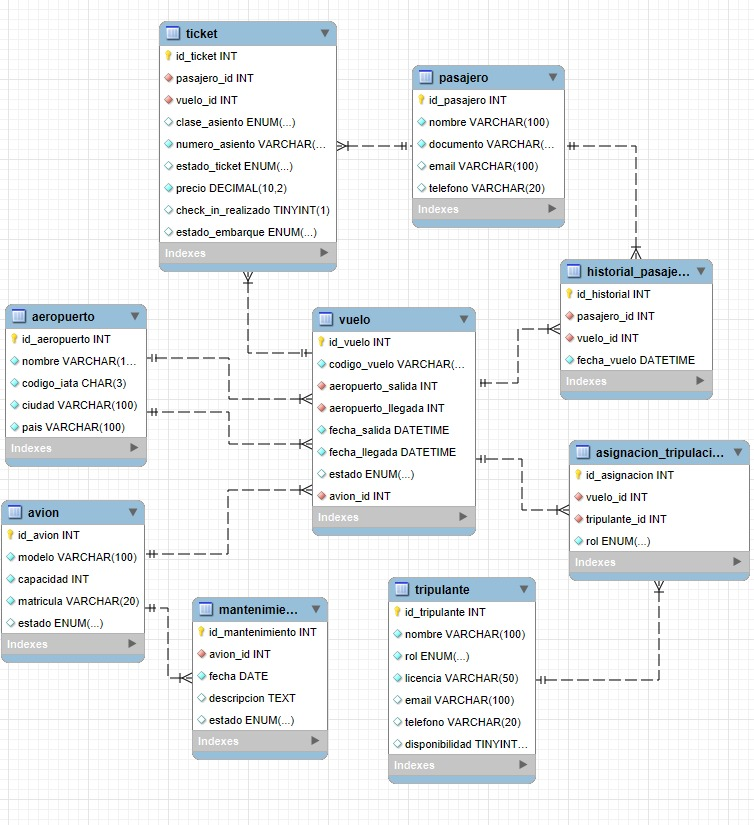
* Diagrama entidad-relación: Representa las entidades principales como Aeropuerto, Avión, Pasajero, Vuelo, Ticket, Tripulante, Asignación de Tripulación, Mantenimiento, Historial de Pasajero y Log de Actividad, y sus relaciones.



* Explicación de entidades principales y relaciones
* Aeropuerto: Contiene información sobre los aeropuertos.
* Avión: Detalles de los aviones y su estado.
* Pasajero: Información de los pasajeros.
* Vuelo: Datos de los vuelos programados.
* Ticket: Información de los boletos emitidos.
* Tripulante: Datos de la tripulación.
* Asignación **de Tripulación**: Relación entre vuelos y tripulantes.
* Mantenimiento: Registro de mantenimientos de aviones.
* Historial **de Pasajero**: Historial de vuelos de los pasajeros.
* Log **de Actividad**: Registro de actividades en la base de datos.

**3.2 Modelo Lógico**

* Diagramas de tablas con atributos y claves



**3.3 Modelo Físico**

* Definición de tablas en SQL

tabla: aeropuerto

create table aeropuerto (

id\_aeropuerto int auto\_increment primary key,

nombre varchar(100) not null,

codigo\_iata char(3) not null unique,

ciudad varchar(100) not null,

pais varchar(100) not null

);

tabla: avión

create table avion (

id\_avion int auto\_increment primary key,

modelo varchar(100) not null,

capacidad int not null,

matricula varchar(20) not null unique,

estado enum('en servicio', 'mantenimiento', 'fuera de servicio') default 'en servicio'

);

tabla: pasajero

create table pasajero (

id\_pasajero int auto\_increment primary key,

nombre varchar(100) not null,

documento varchar(20) not null unique,

email varchar(100),

telefono varchar(20)

);

tabla: logs de actividad

create table log\_actividad (

id\_log int auto\_increment primary key,

usuario varchar(100) not null,

accion varchar(255) not null,

tabla\_afectada varchar(100),

fecha\_hora datetime default current\_timestamp

);

tabla: vuelo

create table vuelo (

id\_vuelo int auto\_increment primary key,

codigo\_vuelo varchar(20) not null unique,

aeropuerto\_salida int not null,

aeropuerto\_llegada int not null,

fecha\_salida datetime not null,

fecha\_llegada datetime not null,

estado enum('programado', 'en vuelo', 'cancelado', 'completado') default 'programado',

avion\_id int not null,

foreign key (aeropuerto\_salida) references aeropuerto(id\_aeropuerto) on delete cascade,

foreign key (aeropuerto\_llegada) references aeropuerto(id\_aeropuerto) on delete cascade,

foreign key (avion\_id) references avion(id\_avion) on delete cascade

);

tabla: ticket

create table ticket (

id\_ticket int auto\_increment primary key,

pasajero\_id int not null,

vuelo\_id int not null,

clase\_asiento enum('economica', 'ejecutiva') default 'economica',

numero\_asiento varchar(10) not null,

estado\_ticket enum('activo', 'cancelado') default 'activo',

precio decimal(10, 2) not null,

check\_in\_realizado boolean default false,

estado\_embarque enum('pendiente', 'completado') default 'pendiente',

foreign key (pasajero\_id) references pasajero(id\_pasajero) on delete cascade,

foreign key (vuelo\_id) references vuelo(id\_vuelo) on delete cascade

);

tabla: historial de pasajero

create table historial\_pasajero (

id\_historial int auto\_increment primary key,

pasajero\_id int not null,

vuelo\_id int not null,

fecha\_vuelo datetime not null,

foreign key (pasajero\_id) references pasajero(id\_pasajero) on delete cascade,

foreign key (vuelo\_id) references vuelo(id\_vuelo) on delete cascade

);

tabla: tripulante

create table tripulante (

id\_tripulante int auto\_increment primary key,

nombre varchar(100) not null,

rol enum('piloto', 'copiloto', 'azafata') not null,

licencia varchar(50) not null unique,

email varchar(100),

telefono varchar(20),

disponibilidad boolean default true

);

tabla: asignación de tripulación

create table asignacion\_tripulacion (

id\_asignacion int auto\_increment primary key,

vuelo\_id int not null,

tripulante\_id int not null,

rol enum('piloto', 'copiloto', 'azafata') not null,

foreign key (vuelo\_id) references vuelo(id\_vuelo) on delete cascade,

foreign key (tripulante\_id) references tripulante(id\_tripulante) on delete cascade

);

tabla: mantenimiento

create table mantenimiento (

id\_mantenimiento int auto\_increment primary key,

avion\_id int not null,

fecha date not null,

descripcion text,

estado enum('pendiente', 'en proceso', 'completado') default 'pendiente',

foreign key (avion\_id) references avion(id\_avion) on delete cascade

);

**4. Diccionario de Datos**

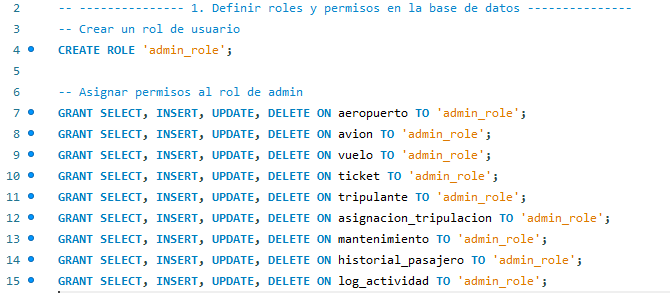
Explicación detallada de cada tabla con:

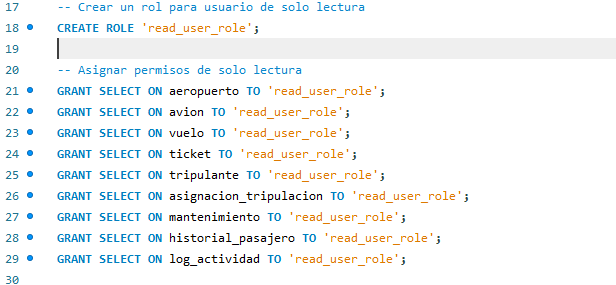
* Nombre de la tabla
* Descripción de la tabla
* Campos (nombre, tipo de datos, restricciones)
* Relación con otras tablas



**5. Seguridad, Auditoría y Control de Acceso**

* Creación de usuarios y roles en la base de datos: Se crean roles como “admin\_role” y “read\_user\_role” con permisos específicos.
* Políticas de acceso: Se definen permisos de lectura, escritura, actualización y eliminación para cada rol.
  + Se hace uso de la base valvoline para la creación de los diferentes permisos.

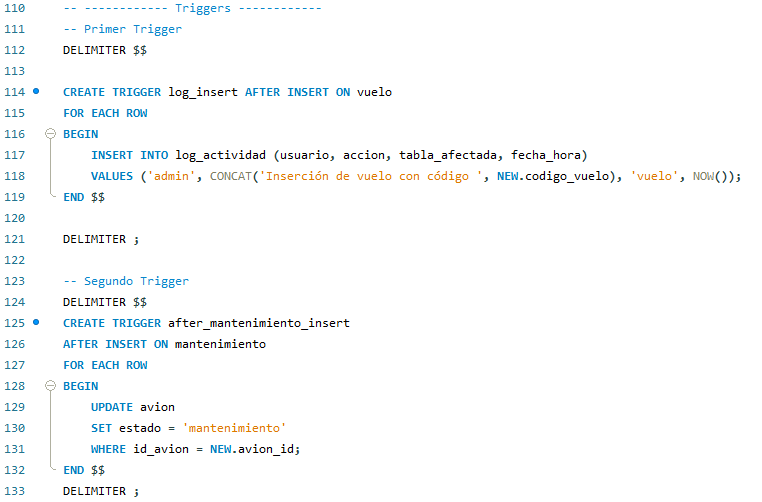




* Cifrado de datos sensibles: Se utiliza AES para cifrar contraseñas de los pasajeros.



* Auditoría y registro de eventos: Se implementan triggers para registrar actividades en la tabla log\_actividad.



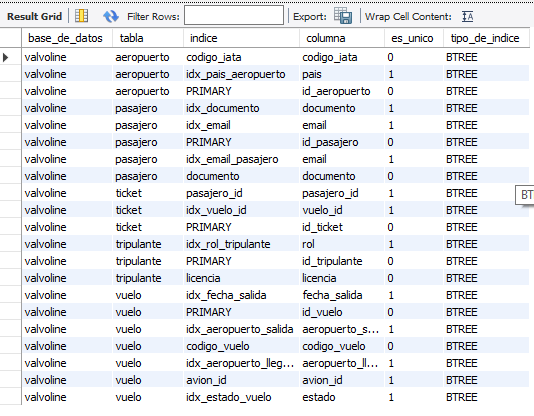
**6. Respaldos y Recuperación de Datos**

* Estrategias de respaldos completos, incrementales y en caliente: Se detallan las estrategias para asegurar la disponibilidad y recuperación de datos.
  + Se implementaron diversas estrategias de respaldo para garantizar la disponibilidad y la integridad de la información en caso de fallos o pérdidas de datos. Entre las técnicas aplicadas se incluyen respaldos completos, incrementales y en caliente, lo que permite una recuperación eficiente en función de las necesidades específicas del sistema.
    - Respaldos Completos: Consisten en la copia íntegra de la base de datos, asegurando que todos los datos estén disponibles para una restauración completa en cualquier momento.
    - Respaldos Incrementales: Permiten almacenar únicamente los cambios realizados desde el último respaldo, optimizando el uso del espacio de almacenamiento y reduciendo el tiempo de respaldo.
    - Respaldos en Caliente: Se realizaron mientras la base de datos permanecía en funcionamiento, lo que minimiza el tiempo de inactividad y asegura la continuidad de las operaciones críticas.

Para la gestión de estos respaldos se utilizó MySQL Workbench, una herramienta que facilita la administración de bases de datos mediante una interfaz gráfica intuitiva. Con Workbench se ejecutaron procesos de exportación de bases de datos, permitiendo la generación de archivos de respaldo. Asimismo, se realizaron pruebas de recuperación para verificar la integridad de los datos y la efectividad de los procedimientos de restauración.

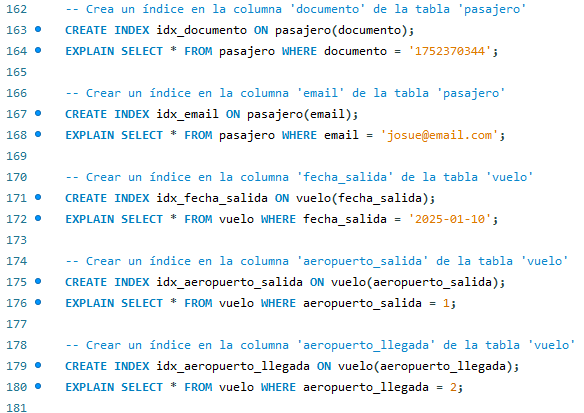
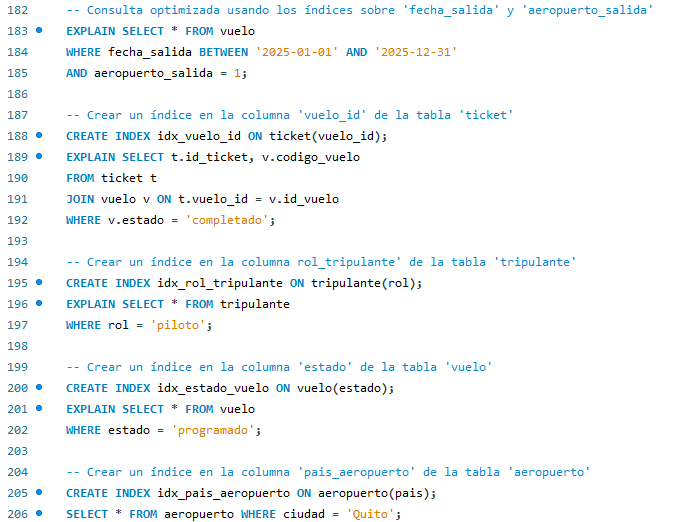
**7. Optimización y Rendimiento de Consultas**

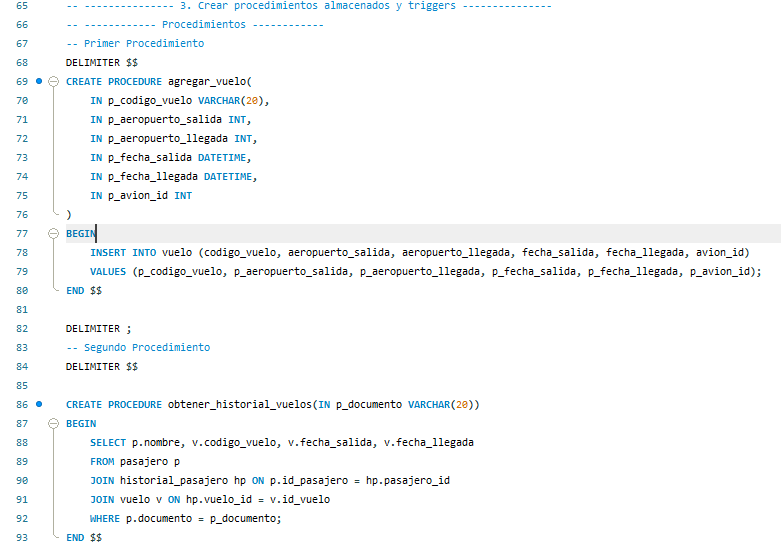
* Uso de índices: Se crean índices en columnas clave para mejorar el rendimiento de las consultas.
* Particionamiento de tablas: Se considera el particionamiento de tablas para mejorar el rendimiento en grandes volúmenes de datos.



* Optimización de consultas con EXPLAIN: Se utiliza EXPLAIN para analizar y optimizar consultas.

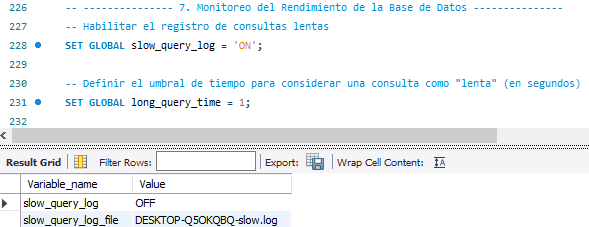
**8. Procedimientos Almacenados, Vistas y Triggers**

Descripción y ejemplos de cada uno: Se incluyen procedimientos almacenados para agregar vuelos, obtener historial de vuelos y actualizar el estado de vuelos. También se implementan triggers para registrar actividades y actualizar el estado de aviones en mantenimiento.



**9. Monitoreo y Optimización de Recursos**

* Métodos de monitoreo y pruebas de carga: Se describen métodos para monitorear el rendimiento de la base de datos y realizar pruebas de carga.





**10. Control de Versiones con Git**

* Uso de Git para la gestión del código SQL: Se utiliza Git para gestionar y versionar los scripts SQL.
* Prácticas recomendadas: Se detallan prácticas recomendadas para el uso de Git en proyectos de bases de datos.

**11. Conclusiones y Recomendaciones**

* Conclusiones: Se resumen los logros del proyecto y su impacto en la gestión de la aerolínea.
* Recomendaciones: Se sugieren mejoras y futuras implementaciones para optimizar aún más el sistema de bases de datos.

**6. Responsabilidades del Equipo - Archivo en .xlsx o .docx**

Debe detallar:

* Miembro responsable
* Tarea asignada
* Fechas de asignación y culminación

|  |  |  |
| --- | --- | --- |
| Responsable | Actividad | Sprint |
| Adrian Ramos | Modelado de datos, diccionario | Sprin1 |
| Josué Guerra | Seguridad y procedimientos | Sprin2 |
| Henry Tonato | Monitoreo y optimización | Sprin3 |
| Adrian Ramos | Documentación y presentación | Sprin4 |